1. Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median.

library(ISLR)

## Warning: package 'ISLR' was built under R version 3.2.5

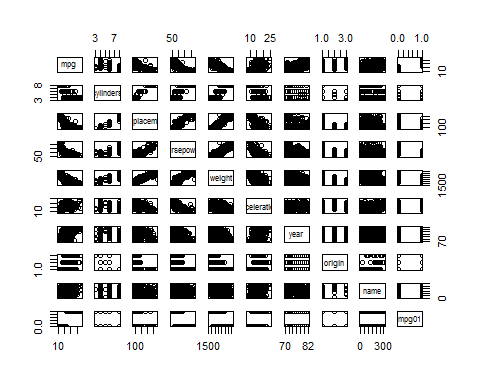
rm(Auto)

## Warning in rm(Auto): object 'Auto' not found

attach(Auto)  
Auto$mpg01 <- 0  
Auto[mpg > median(mpg),]$mpg01 <- 1

(b)Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01?

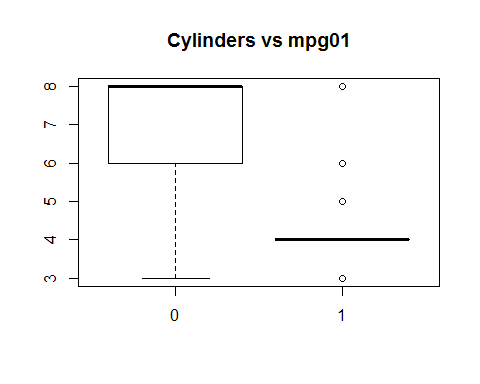
pairs(Auto)



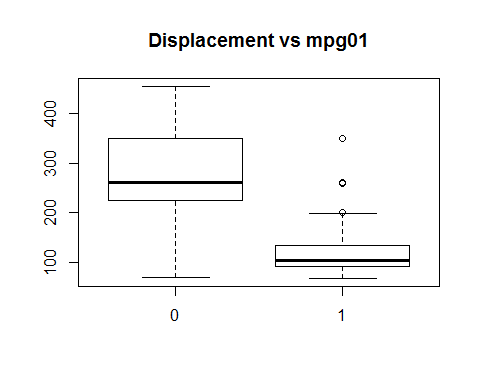
cor(Auto[,-9])

## mpg cylinders displacement horsepower weight  
## mpg 1.0000000 -0.7776175 -0.8051269 -0.7784268 -0.8322442  
## cylinders -0.7776175 1.0000000 0.9508233 0.8429834 0.8975273  
## displacement -0.8051269 0.9508233 1.0000000 0.8972570 0.9329944  
## horsepower -0.7784268 0.8429834 0.8972570 1.0000000 0.8645377  
## weight -0.8322442 0.8975273 0.9329944 0.8645377 1.0000000  
## acceleration 0.4233285 -0.5046834 -0.5438005 -0.6891955 -0.4168392  
## year 0.5805410 -0.3456474 -0.3698552 -0.4163615 -0.3091199  
## origin 0.5652088 -0.5689316 -0.6145351 -0.4551715 -0.5850054  
## mpg01 0.8369392 -0.7591939 -0.7534766 -0.6670526 -0.7577566  
## acceleration year origin mpg01  
## mpg 0.4233285 0.5805410 0.5652088 0.8369392  
## cylinders -0.5046834 -0.3456474 -0.5689316 -0.7591939  
## displacement -0.5438005 -0.3698552 -0.6145351 -0.7534766  
## horsepower -0.6891955 -0.4163615 -0.4551715 -0.6670526  
## weight -0.4168392 -0.3091199 -0.5850054 -0.7577566  
## acceleration 1.0000000 0.2903161 0.2127458 0.3468215  
## year 0.2903161 1.0000000 0.1815277 0.4299042  
## origin 0.2127458 0.1815277 1.0000000 0.5136984  
## mpg01 0.3468215 0.4299042 0.5136984 1.0000000

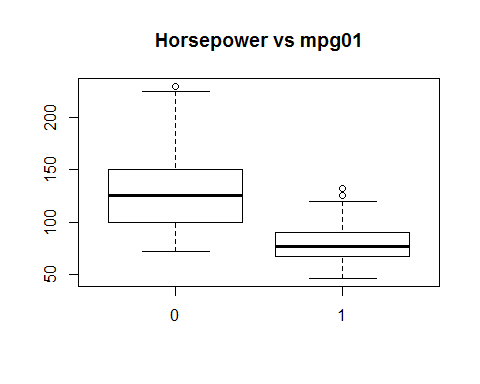
boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders vs mpg01")



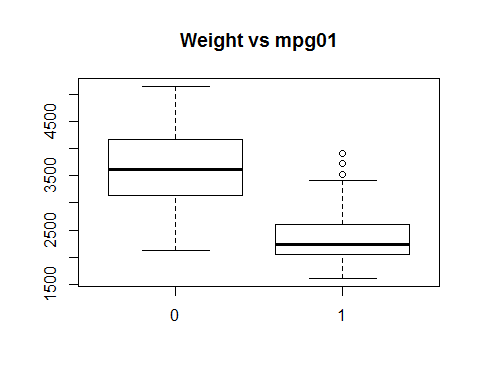
boxplot(displacement ~ mpg01, data = Auto, main = "Displacement vs mpg01")



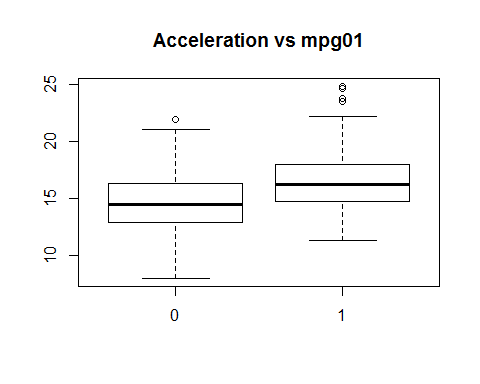
boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower vs mpg01")



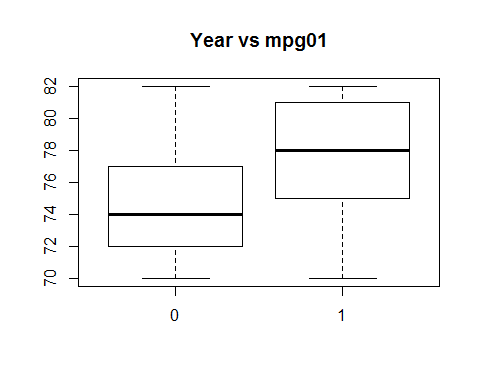
boxplot(weight ~ mpg01, data = Auto, main = "Weight vs mpg01")



boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration vs mpg01")



boxplot(year ~ mpg01, data = Auto, main = "Year vs mpg01")



1. Split the data into a training set and a test set.

## add rnum coloumn  
Auto$rnum<-seq(1,nrow(Auto),1)  
## split data  
Auto\_train <- Auto[Auto$rnum %% 2 ==0, ]  
Auto\_test<- Auto[Auto$rnum %% 2 !=0, ]  
##drop runm coloumn  
Auto\_train<-Auto\_train[,!(names(Auto\_train) %in% c("rnum"))]  
Auto\_test<-Auto\_test[,!(names(Auto\_test) %in% c("rnum"))]  
Auto<-Auto[,!(names(Auto) %in% c("rnum"))]

(d)Perform LDA on the training data in order to predict "mpg01" using the variables that seemed most associated with "mpg01" in (b). What is the test error of the model obtained ?

library(MASS)  
lda\_model <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto\_train)  
lda\_model

## Call:  
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.4897959 0.5102041   
##   
## Group means:  
## cylinders weight displacement horsepower  
## 0 6.760417 3653.583 273.500 132.4271  
## 1 4.170000 2305.070 114.045 78.8700  
##   
## Coefficients of linear discriminants:  
## LD1  
## cylinders -0.4297440160  
## weight -0.0011996694  
## displacement 0.0003516146  
## horsepower 0.0021885992

lda\_pred <- predict(lda\_model, Auto\_test)  
table(lda\_pred$class, Auto\_test$mpg01)

##   
## 0 1  
## 0 83 6  
## 1 17 90

mean(lda\_pred$class !=Auto\_test$mpg01)

## [1] 0.1173469

Hence the error rate is

(e)Perform QDA on the training data in order to predict "mpg01"

qda\_model <- qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto\_train)  
qda\_model

## Call:  
## qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto\_train)  
##   
## Prior probabilities of groups:  
## 0 1   
## 0.4897959 0.5102041   
##   
## Group means:  
## cylinders weight displacement horsepower  
## 0 6.760417 3653.583 273.500 132.4271  
## 1 4.170000 2305.070 114.045 78.8700

qda\_pred <- predict(qda\_model, Auto\_test)  
table(qda\_pred$class, Auto\_test$mpg01)

##   
## 0 1  
## 0 89 9  
## 1 11 87

mean(qda\_pred$class != Auto\_test$mpg01)

## [1] 0.1020408

Hence the error rate with QDA model is which is higher/lower compared to LDA

(f)Perform logistic regression on the training data in order to predict "mpg01"

glm\_model <- glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto\_train, family = binomial)  
summary(glm\_model)

##   
## Call:  
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,   
## family = binomial, data = Auto\_train)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.2416 -0.1088 0.1038 0.3176 2.9731   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 13.1401211 2.5218930 5.210 1.88e-07 \*\*\*  
## cylinders -0.0665897 0.4846865 -0.137 0.8907   
## weight -0.0025557 0.0009355 -2.732 0.0063 \*\*   
## displacement -0.0084554 0.0109024 -0.776 0.4380   
## horsepower -0.0431963 0.0197508 -2.187 0.0287 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 271.632 on 195 degrees of freedom  
## Residual deviance: 97.301 on 191 degrees of freedom  
## AIC: 107.3  
##   
## Number of Fisher Scoring iterations: 7

ods <- predict(glm\_model, Auto\_test, type = "response")  
glm\_pred <- rep(0, length(ods))  
glm\_pred[ods > 0.5] <- 1  
table(glm\_pred, Auto\_test$mpg01)

##   
## glm\_pred 0 1  
## 0 88 8  
## 1 12 88

mean(glm\_pred != Auto\_test$mpg01)

## [1] 0.1020408

Test Error rate here is:

(g)Perform KNN on the training data, with several values of KK, in order to predict "mpg01"

set.seed(12)  
library(class)

## Warning: package 'class' was built under R version 3.2.5

## with K =5  
knn\_pred <- knn(Auto\_train[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_test[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_train$mpg01, k = 5)  
table(knn\_pred, Auto\_test$mpg01)

##   
## knn\_pred 0 1  
## 0 86 17  
## 1 14 79

mean(knn\_pred != Auto\_test$mpg01)

## [1] 0.1581633

## with K =10  
knn\_pred <- knn(Auto\_train[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_test[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_train$mpg01, k = 10)  
table(knn\_pred, Auto\_test$mpg01)

##   
## knn\_pred 0 1  
## 0 85 18  
## 1 15 78

mean(knn\_pred != Auto\_test$mpg01)

## [1] 0.1683673

## with K =100  
knn\_pred <- knn(Auto\_train[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_test[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_train$mpg01, k = 100)  
table(knn\_pred, Auto\_test$mpg01)

##   
## knn\_pred 0 1  
## 0 82 7  
## 1 18 89

mean(knn\_pred != Auto\_test$mpg01)

## [1] 0.127551

## with K =125  
knn\_pred <- knn(Auto\_train[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_test[,c("cylinders", "weight", "displacement", "horsepower")], Auto\_train$mpg01, k =125)  
table(knn\_pred, Auto\_test$mpg01)

##   
## knn\_pred 0 1  
## 0 78 4  
## 1 22 92

mean(knn\_pred != Auto\_test$mpg01)

## [1] 0.1326531

For KNN with K=125, the error rate is lower, hence better performace model